

A Trajectory-based Calibration Method for Stochastic Motion Models

Ezequiel Di Mario, Grégory Mermoud, Massimo Mastrangeli and Alcherio Martinoli

Abstract—In this paper, we present a quantitative, trajectory-based method for calibrating stochastic motion models of water-floating robots. Our calibration method is based on the Correlated Random Walk (CRW) model, and consists in minimizing the Kolmogorov-Smirnov (KS) distance between the step length and step angle distributions of real and simulated trajectories generated by the robots. First, we validate this method by calibrating a physics-based motion model of a single 3-cm-sized robot floating at a water/air interface under fluidic agitation. Second, we extend the focus of our work to multi-robot systems by performing a sensitivity analysis of our stochastic motion model in the context of Self-Assembly (SA). In particular, we compare in simulation the effect of perturbing the calibrated parameters on the predicted distributions of self-assembled structures. More generally, we show that the SA of water-floating robots is very sensitive to even small variations of the underlying physical parameters, thus requiring real-time tracking of its dynamics.

I. INTRODUCTION

When it comes to characterizing the trajectory of a robot, it is often practical to use one of these opposed assumptions: (i) the robot follows a deterministic trajectory defined by a given control law, e.g. [1]; or (ii) the robot performs a random walk with some known average speed that can be mapped to some diffusion coefficient, e.g. [2]. However, there are many situations, such as the one presented in this paper, where the reality lies between these two extremes. In our case, we study the stochastic motion of 3-cm-sized water-floating cubic robots, denoted as *blocks* hereafter, within a tank. These blocks weight approximately 11.6 g, and are endowed with a visual marker for tracking purposes, as well as with four permanent magnets (one on each side) for mutual aggregation (see Fig. 1b). The blocks are not self-locomoted; instead, they are stirred by the fluid flow produced by pumps along the tank perimeter. As a result, they describe trajectories with well-defined geometric features, yet with a strong stochastic component.

In this work, we first study the calibration of the motion model based on a single block's trajectory, and we then investigate in simulation the impact of this motion model in a multi-robot scenario. In particular, we use the very same motion model in a physics-based simulation of self-assembling blocks, and we observe the effect of perturbing the calibrated parameters on assembled structures. In other

words, we show how important a proper calibration of the motion model is when modeling Self-Assembly (SA) processes.

SA is the autonomous organization of pre-existing physical units into spatial patterns or structures [3]. Fluidic SA has attracted increasing interest as a competitive alternative to traditional manufacturing techniques for micro and nano electro-mechanical systems [4], [5]. Still, it presents a number of methodological and experimental challenges. For instance, a broad range of interactions have to be measured and modeled (e.g., hydrophobic, capillary, electrostatic and magnetic forces), suitable agitation methods have to be determined, and an appropriate imaging technique has to be developed to assess the assembly progress and yield. Working at a larger scale may alleviate some of these difficulties, especially those concerning block fabrication and manipulation, while still allowing to gain consistent insights on the SA process. Therefore, several platforms at larger scales are being used to study SA [6]–[12], exploiting similarities with sub-millimeter scale SA without dealing with all the difficulties that arise from extreme miniaturization. These platforms present different physical dimensions, degrees of autonomy, and computational capabilities. For instance, in [12] the blocks are completely passive, which allows for reduced block sizes in fluidic environments, but the control of the SA process relies entirely on an active patterned substrate. In [9] and [11] there is some kind of electronic control and actuators on board, though energy is provided through inter-block physical connections and is therefore not available to the blocks until they latch to the main aggregate. Miyashita et al. [10] address the block energy autonomy problem by using an external pantograph to draw power from the top of the tank; however, the only control parameter used in this system is the globally applied power. Klavins' Programmable Parts [7] could be considered fully autonomous self-assembling robots, only lacking self-locomotion and relying on an air table for agitation. Nevertheless, it is not clear if results obtained from this setup can be extended to fluidic SA at smaller scales, due to the parts' relatively large size and the agitation method not including the peculiar fluidic interactions.

Comparatively little effort has been devoted to modeling robotic SA [13]. Models with high level of abstraction are usually non-spatial and assume well-mixed systems; yet, Napp et al. [14] have shown that, in their Programmable Parts setup, the robots are more likely to interact with their last interaction partner than with any other. In fact, the implications of these assumptions are difficult to gauge due to the lack of appropriate modeling tools for the previously mentioned

The authors are with the Distributed Intelligent Systems and Algorithms Laboratory (DISAL), School of Architecture, Civil and Environmental Engineering, École Polytechnique Fédérale de Lausanne. Ezequiel Di Mario is sponsored by the Swiss National Science Foundation through the National Center of Competence in Research Robotics. Grégory Mermoud and Massimo Mastrangeli are sponsored by the Nano-Tera.ch research initiative in the framework of the SelfSys project. ezequiel.dimario@epfl.ch

robotic platforms. State-of-the-art robotic simulators such as Webots or Gazebo do not support fluid dynamics natively. Moreover, to our knowledge little importance has been given to real-time tracking of block trajectories and their impact on the dynamics and the outcome of SA processes, which we intend to show to be non-negligible.

The paper is organized as follows: Section II describes the experimental setup used to collect the trajectories; Section III introduces a motion model of the water-floating block accounting for simple hydrodynamic effects such as drag, buoyancy, and, to some extent, fluctuations due to stirring; Section IV describes the main contribution of the paper, that is, a novel method to calibrate and validate the parameters of such models based on the comparison between the simulated and experimental trajectories of the blocks. Finally, Section V-A discusses the experimental results of the calibration procedure; and Section V-B reports the results of our sensitivity analysis, performed in simulation.

II. EXPERIMENTAL SETUP

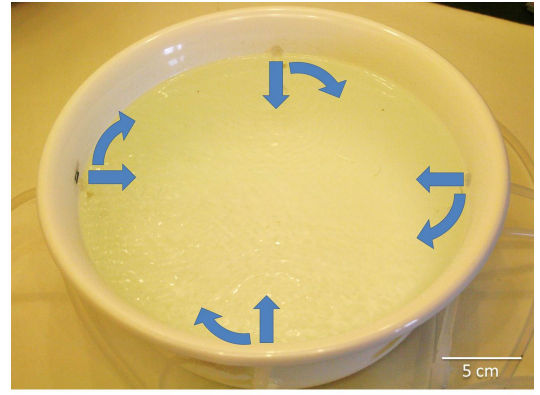
The experimental setup consists of a circular water tank of approximately 30 cm in diameter, with eight inlets connected to four diaphragm pumps. Four inlets are perpendicular to the wall, and the other four are as tangential as possible, allowing to create both radial and circular flows. Additionally, four outlets are placed at the bottom of the tank, to avoid affecting the flow at the surface as much as possible. Each pump's flow rate can be controlled individually up to a maximal value of 600 ml/min. This flexible configuration allows us to investigate a variety of different flow patterns and associated block trajectories. Indeed, perpendicular inlets generate very irregular trajectories, and allow for collisions in the middle of the tank, but they exhibit dead spots near the walls. Conversely, tangential inlets generate circular flows that prevent dead spots, but lead to very regular, closed trajectories that do not favor collisions.

To monitor the evolution of the system in real time, we use an overhead camera to track the pose of the blocks. A two-color passive marker located at the top of each block is used in conjunction with SwisTrack [15], an open-source software package developed in our laboratory, to track the blocks. Both their position (x, y) and orientation (θ) are logged in real time at a rate of approximately 10 Hz.

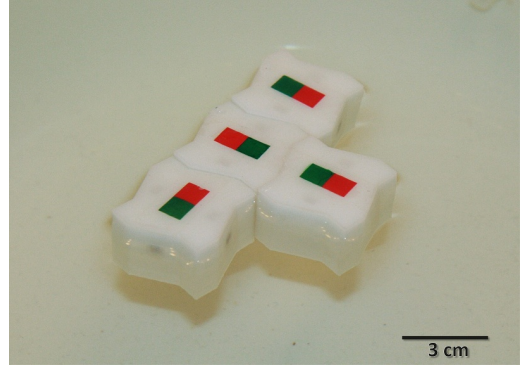
III. MOTION MODEL

To reproduce the distributed system in simulation, we use Webots [16], a mobile, physics-based robotics simulation software. Webots uses the Open Dynamics Engine (ODE) for simulating rigid body dynamics, but it does not support fluidic dynamics. However, in order to accurately simulate our system, we need to capture both the flow generated by the pumps and the interactions between the robots and the flow.

Our first approach to solve this problem was to couple Webots with a Computational Fluid Dynamics (CFD) tool, for instance the Lattice Boltzmann method. However, this



(a)



(b)

Fig. 1: Experimental setup. (a) Tank with 8 inlets (4 perpendicular and 4 tangential to the wall). (b) Three assembled blocks floating on water.

approach is not only computationally expensive, but also difficult to implement in order to ensure numerical stability of the simulation. Also, our system being essentially stochastic and subject to important asymmetries and manufacturing imperfections that may significantly affect the flow, an approach based on CFD tools does not represent a suitable choice.

As a result, we decided to adopt a completely different approach. Instead of trying to predict the flow resulting from a certain pump configuration, we use the blocks' trajectories to extract the flow velocity field, and then use it in the Webots simulations. This approach dramatically decreases the computational cost as compared to CFD methods, but it requires that the flow velocity field for each possible pump configuration be measured beforehand.

Concretely, we record the trajectory of a single, floating cubic block (with side length of 3 cm) during 30 minutes, and then construct a discrete regular velocity field by discretizing the trajectory plane into a regular grid, and averaging the observed velocity vectors at each cell of the grid. Choosing the number of divisions of the grid is a compromise similar to choosing the number of bins in a histogram: a too coarse grid may hide important features of the data, while a too fine grid could result in very few samples per division, and therefore less statistical significance. We used a discretization of about 50 cells in each dimension for our arena (30 cm in diameter).

Once the velocity field is extracted, a correction is applied

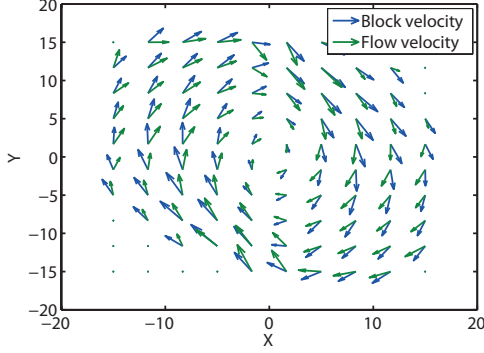


Fig. 2: Velocity field of the tracked block and calculated flow.

to compensate for the inertia of the block. This correction is based on a simple dynamical analysis considering the drag force, which is the only force acting in the horizontal plane. First, we compute the Reynolds number Re , which determines the flow regime of our system:

$$Re = \frac{\rho \cdot V \cdot L}{\mu} \sim 2000 \quad (1)$$

where $\rho = 10^3 \text{ kg/m}^3$ is the density of water, $V \sim 6.4 \text{ cm/s}$ the experimentally-measured mean velocity of the object relative to the fluid, $L = 3 \text{ cm}$ the characteristic block linear dimension, and $\mu = 8.90 \cdot 10^{-4} \text{ Ps}\cdot\text{s}$ the dynamic viscosity of water. This value of Re is quite higher than the typical values ($Re < 10$) present at smaller scales [17], in which the linear drag Stokes' law is usually applied. In our case, this high Reynolds number indicates that the drag force takes a quadratic form:

$$|\vec{F}_{drag}| = \frac{1}{2} \cdot \rho \cdot A \cdot C_x \cdot |\vec{v}_{block} - \vec{v}_{flow}|^2 \quad (2)$$

where \vec{v}_{block} is the block's velocity, \vec{v}_{flow} is the flow's velocity, A the block's cross sectional area to the flow, and C_x a dimensionless drag coefficient.

The direction of the drag force is opposed to the velocity of the block relative to the flow, as stated by:

$$\frac{\vec{F}_{drag}}{|\vec{F}_{drag}|} = -\frac{\vec{v}_{block} - \vec{v}_{flow}}{|\vec{v}_{block} - \vec{v}_{flow}|} \quad (3)$$

Assuming $A = 9 \text{ cm}^2$ (i.e., the area of one block's side, since the block is almost completely immersed) and the drag coefficient of a cube $C_x = 1.05$, and taking into account that the mass of the block can be measured, and its velocity and acceleration can be obtained from the tracked trajectories, the only unknowns are the x and y components of the flow velocity, which result in the following equations:

$$v_{flow,i} = v_{block,i} + \frac{m \cdot a_i}{\sqrt{\frac{1}{2} \cdot \rho \cdot A \cdot C_x \cdot m \sqrt{a_x^2 + a_y^2}}}, \quad i = x, y \quad (4)$$

It must be noted in Fig. 2 that the computed flow velocity has a centripetal component that generates the observed circular trajectory. If the correction of Equation 4 is not applied, the block constantly crashes onto the walls, which

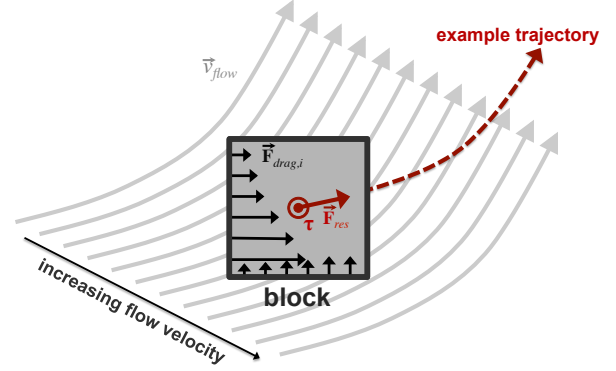


Fig. 3: Torque τ and force \mathbf{F}_{res} on a block in a non-uniform fluid flow.

is not consistent with experimental observations. Note that, importantly, our simulation is completely independent from the method used for measuring the flow velocity; in this particular case, we used the same object for measuring the flow velocity and in our simulation, but this needs not to be the case in principle. For example, Particle Image Velocimetry (PIV) could also be used to determine the flow velocity field.

Rigid body dynamics is not sufficient to simulate the motion of the blocks given the flow velocity. To solve this problem, we developed a physics plugin for Webots; this plugin applies the Archimedes' force, which is simply the weight of the fluid displaced by the block, and the drag force based on the flow velocity at the location of the block. However, here the drag is not computed once for the whole block and added to its center of mass; instead, the drag force is integrated over each face of the block to account for rotational effects. To do this, each face is divided into N planes, and the drag force is computed for each plane according to Eq. 2 using appropriate values of A and C_x . On the one hand, the area A becomes the plane's cross sectional area to the flow (which depends on the plane's orientation relative to the flow). On the other hand, C_x is now the drag coefficient for each of the small planes. This coefficient is unknown and becomes a free parameter to be calibrated, which we denote C_d for the sake of clarity. The calculated force is not added to the block's center of mass but to the center of each plane, which allows for capturing the vorticity of the flow, e.g., the block will rotate when faced with a flow whose strength is increasing across one face, as shown in Fig. 3.

Importantly, the physics plugin also adds a stochastic force F_{stoch} to the center of mass of each block in order to account for non-modeled effects (e.g., physical irregularities, turbulences). F_{stoch} is gaussian with zero mean and standard deviation σ_F . As a result, the simulation has two parameters to be calibrated: the dimensionless drag coefficient C_d and the standard deviation σ_F of the stochastic force.

IV. CALIBRATION METHOD

The basic principle of our calibration method is to compare *simulated* and *real* trajectories so that the actual behavior of the blocks is reproduced as faithfully as possible. Sampled

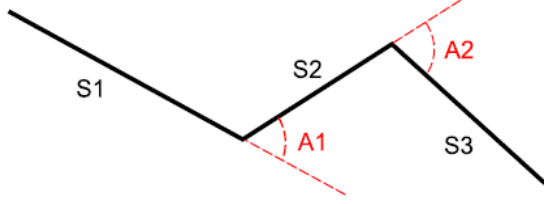


Fig. 4: Step lengths S_i and angles A_i for the Correlated Random Walk (CRW) model.

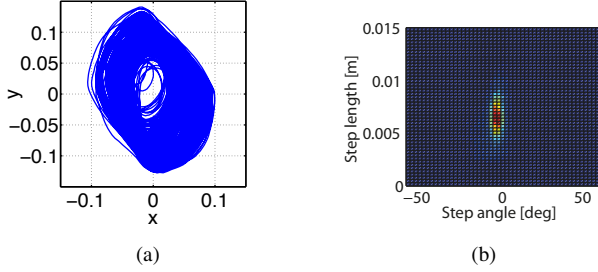


Fig. 5: (a) A sample trajectory. (b) Its step length and angle distributions.

trajectories can be viewed as sequences of points in a two-dimensional space: there is no need to consider the temporal dimension as we are using a constant sampling rate. In our setup, real and simulated trajectories do not have a common frame because there are no reference points or pre-established paths, just the blocks reacting to the environment. Therefore, it is not appropriate to compare trajectories in the Euclidean space because, even though they are generated by the same agitation mechanism and should therefore exhibit common traits, the sequence of points will vary greatly depending on initial conditions and random collisions with walls. A solution described in [18] is to use the Correlated Random Walk (CRW) model, which consists in constructing the trajectory's bi-dimensional distribution of the step length and step angle (Fig. 4). A step is defined as the segment that connects two points of the trajectory, sampled at time t and $t + \Delta t$, respectively. A sample trajectory and its corresponding step length and angle distributions can be seen in Fig. 5.

In order to quantitatively compare real and simulated trajectories in the bi-dimensional space spanned by step lengths and step angles, we use the Kolmogorov-Smirnov (KS) distance, that is, the maximum distance between the corresponding cumulative distribution functions. The reasoning behind this is that the KS statistical test is non-parametric (i.e., no assumption is made about the underlying distributions), and it is sensitive to both the shape and the location of the empirical distribution functions. Therefore, our calibration method does not only try to match the mean linear and angular velocity of the real trajectory, but also the shape of the bi-dimensional distribution as a whole.

When extending the KS test to two dimensions, the calculation of the cumulative distribution function is not as straightforward as in the one-dimensional case, as there are four possible ways to cumulate data following the directions of the coordinate axes. In our implementation, we followed

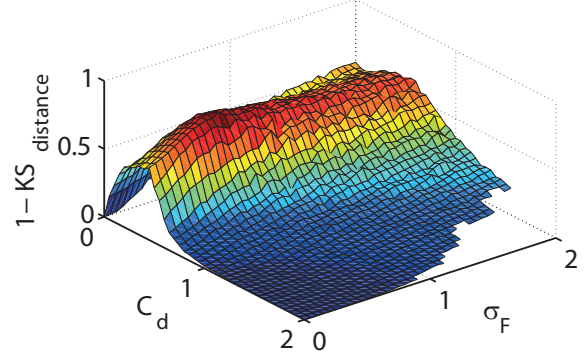


Fig. 6: Plot of $1 - KS_{distance}$ between real and simulated trajectories as a function of the drag coefficient C_d and the standard deviation σ_F of the added stochastic force. Missing data in the plot are due to numerical instabilities for too large values of C_d and F_{stoch} .

the Fasano and Franceschini variation presented in [19], which has been shown to reduce the computational complexity from $O(n^3)$ in Peacock's original version to $O(n^2)$ without sacrificing the test's power to distinguish dataset differences [20].

Finally, we systematically explore the parameter space to find the optimal modeling parameters, i.e., those that minimize the KS distance between the simulated and real trajectories. Due to the small number of parameters in our case study, there is no need for more complex optimization techniques. However, more complex optimization methods can be used in principle, as long as they are noise-resistant [21].

V. RESULTS

In Section V-A, we report the results of our calibration method, and we provide both quantitative and qualitative insights into the influence of physical parameters on the trajectory of a *single* block under fluidic agitation. In Section V-B, we investigate in simulation the sensitivity of the SA of *multiple* blocks to the same parameters.

A. Calibration

In our systematic exploration, we vary the drag coefficient C_d between 0 and 2 with increments of 0.05, and the standard deviation of the stochastic force F_{stoch} from 0 mN to 2 mN with increments of 0.05 mN. For each set of parameters, we simulate a trajectory of 30 minutes, and compute the KS distance between the resulting simulated trajectories and a real, pre-recorded trajectory using all pumps at full power, with two pumps directly connected to opposite perpendicular inlets and two other connected to both one perpendicular and one tangential inlet each. All trajectories are sampled at the same rate (10 Hz) in order to be able to directly compare step lengths and angles. Fig. 6 shows a plot of $1 - KS_{distance}$ for improved visualization of the optimum.

The KS distance is minimized for a drag coefficient $C_d = 0.45$, and a standard deviation of the stochastic force $\sigma_F = 0.55$ mN. One can observe a ridge along the drag coefficient axis, which indicates that the faithfulness of the simulation rapidly decreases when leaving the $[0.3 \ 0.5]$ range for the

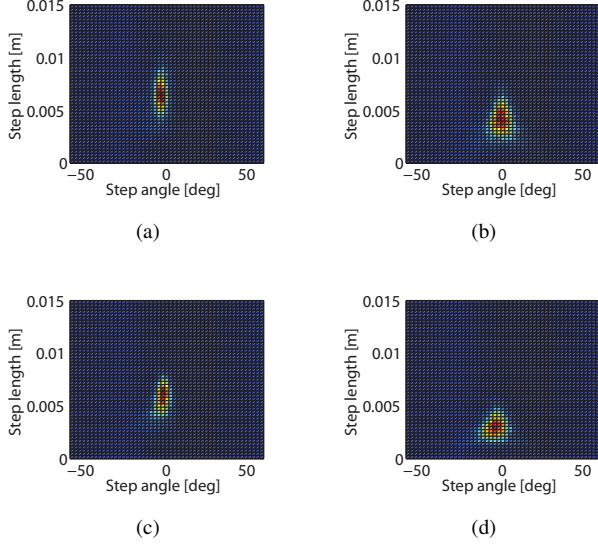


Fig. 7: Step length and angle distributions for different drag coefficients. (a) Real trajectory. (b) $C_d = 0.1$. (c) $C_d = 0.45$. (d) $C_d = 0.8$.

drag coefficient. Values for the stochastic force and the drag coefficient that are too large lead to numerical instabilities.

The qualitative effects of different values for the drag coefficient are shown in Fig. 7. If the drag coefficient is too low, the block does not follow the flow: the mean step length becomes shorter, and the mean angle is around zero, as opposed to the negative step angles of the real trajectory that show the prevailing turning direction of the circular flow. Conversely, very high values of the drag coefficient cause the block to be dragged to the center of the tank, and eventually to rotate almost in place, thereby resulting in very short step lengths and negative step angles due to the rotation. Regarding the stochastic force (Fig. 8), low values of the standard deviation lead to very regular trajectories, and the resulting distribution shows very little variance. However, if the stochastic force is too high, the trajectories become very irregular and similar to those observed in Brownian motion, while the step length and angle distribution becomes sparse.

B. Sensitivity Analysis of Self-Assembly Simulations

Up to now, we only discussed the impact of the simulation's physical parameters on the trajectory of a *single* block in the tank. In this section, we investigate in simulation the sensitivity of the multi-block SA process to these parameters. Each block is endowed with four latching mechanisms (one on each side), which allow the blocks to automatically align and irreversibly attach to each other upon collision, simulating the effects of a magnetic connection.

We compare the outcome of simulated SA experiments with varying values of the drag coefficient C_d and the standard deviation σ_F of the stochastic force. For each set of parameters, we perform 1000 simulation runs (5 minutes of simulated time each), and classify the resulting self-assembled structures into geometrical classes. For the sake of simplicity, we use only four blocks in this study, which leads

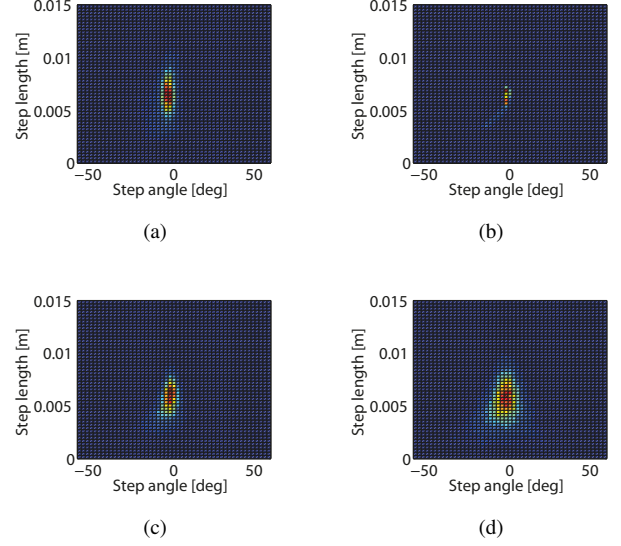


Fig. 8: Step length and angle distributions for different standard deviations of the stochastic force. (a) Real trajectory. (b) $\sigma_F = 0$ mN. (c) $\sigma_F = 0.55$ mN. (d) $\sigma_F = 1$ mN.

to 10 different classes¹ depicted at the bottom of Fig. 11. In the particular case of the optimal parameters found in Section V-A, we perform 10 realizations of 1000 runs to obtain the corresponding distribution, which we will use as reference, and its standard deviation values (Fig. 9).

We then compare the resulting distributions of self-assembled geometries using the $l1$ -norm:

$$dist. = \sum_{i=1}^{10} |x_i^a - x_i^b| \quad (5)$$

where x_i^a , and x_i^b represent the proportion of geometries of class i for the set of parameters a and b , respectively. Fig. 11 depicts the distance (as defined in Eq. 5) between the reference distribution (see Fig. 9) and distributions yielded by parameters ranging between 0 and 1.5 for C_d and 0 mN and 1.5 mN for σ_F .

Fig. 11 clearly shows that the structures resulting from simulated SA are sensitive to variations of both the drag coefficient and the random force. First, one can observe important qualitative variations of the obtained distributions throughout the parameter space. The magnitude of these variations is much larger than the fluctuations observed in different realizations of the reference distribution (see Fig. 9). Second, a more quantitative analysis shows that the most important divergences occur for small values of either parameters. Interestingly, the smallest distance is not achieved in the region closest to the reference distribution; instead, very large stochastic forces lead to distributions that are most similar to the reference distribution. This finding is very important, as it shows that very small variations of the

¹Four blocks with four binding sites produce 12 different geometrical classes, but two pairs of classes (labelled 3 and 5 in Fig. 11) correspond to the same chiral structure, which we do not distinguish in this work.

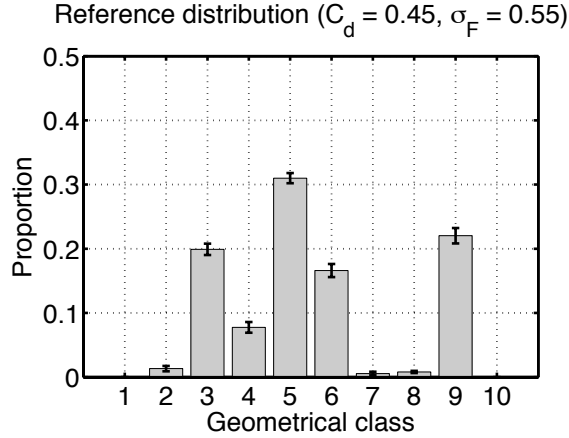


Fig. 9: The average distribution of self-assembled geometries for $C_d = 0.45$ and $\sigma_F = 0.55$ mN (i.e., the optimal parameters reported in Section V-A) averaged over 10 realizations of 1000 runs. The error bars denote the standard deviation, and the average distance (as defined in Eq. 5) between realizations is 0.065.

physical parameters can lead to very important changes in the outcome of the simulation. It also further underlies the importance of a proper parameter calibration. Conversely, we understand that one cannot rely only on the final outcome of a SA experiment to assess the physical properties of the system. While this issue is not crucial at the centimeter-scale, where real-time and accurate tracking of each building block is feasible, it poses important challenges for studying SA at the microscale, where such detailed analysis of the system's dynamics is very difficult.

A Principal Component Analysis (PCA) can help to visualize the variation in the dataset shown in Fig. 11. Fig. 10 depicts both the dataset and the original variables (i.e., the 10 geometrical classes) projected on the 2-dimensional plane spanned by the two principal components. Interestingly, one can directly observe that principal component 1 is aligned with the physically disconnected geometries (classes 1, 2, 8, 9, and 10), which are favored by very small values of C_d and σ_F . Indeed, while disconnected geometries are never observed in most scenarios (namely, when the mobility of the blocks with respect to each other is sufficiently high), they can become very prominent in others (e.g., see the distribution for $C_d = 0.43$ and $\sigma_F = 0.54$ in Fig. 11). On the other hand, the large elongated cluster aligned with the second component corresponds to the scenarios that yield a unique aggregate; interestingly, there is no region of the space that contains most of the points. This observation further confirms that SA is very sensitive to variations of the physical parameters of the model.

VI. CONCLUSION

We introduced a novel method for calibrating models of stochastic SA of floating blocks based on the Correlated Random Walk (CRW) model. In particular, we optimized two physical parameters of our physics-based simulation (i.e., the drag coefficient C_d and the standard deviation σ_F of the stochastic force) by minimizing the Kolmogorov-Smirnov

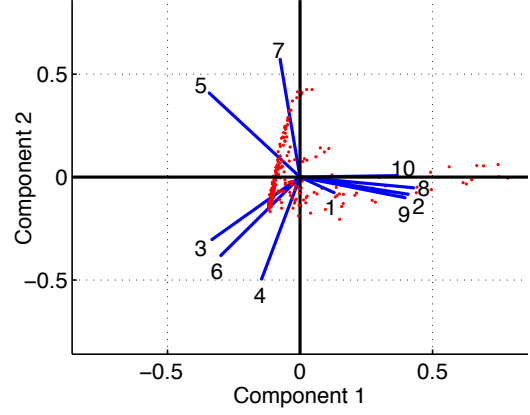


Fig. 10: Biplot of the dataset (red dots) shown in Fig. 11 projected on the 2-dimensional plane spanned by the two principal components. Projections of the original geometrical classes are depicted by blue lines.

(KS) distance between real and simulated trajectories described by the blocks.

In spite of its simplicity, our physical simulation allows for the accurate reproduction, after calibration, of real trajectories of centimeter-sized floating blocks under fluidic agitation. However, the simulation has the main limitation that it relies on a static flow velocity field. It might therefore not be suitable if the agitation pattern is changed over time to enhance stirring, or if the effect of the blocks on the fluid flow cannot be neglected.

We believe that the novelty of our approach lies in the application of trajectory analysis to the study of SA, particularly as a way to study the effect of physical parameters and stirring patterns on the resulting self-assembled structures. For instance, it would be interesting to analyze how the tank's size and different pump configurations favour or restrict the formation of certain geometrical classes. Importantly, the inherent sensitivity of such structures hereby evidenced strongly motivates the need for real-time tracking of SA processes.

This work is part of our ongoing effort to develop a methodological framework for modeling and controlling SA at all scales. While this work assumed the feasibility of tracking individual blocks in a reliable and accurate manner, we envision extending our approach to scenarios where the system is only partially observable, both in time and in space. Another important thread for future research concerns the extension of our calibration method to 3D systems.

VII. ACKNOWLEDGEMENTS

The authors would like to acknowledge Emmanuel Droz for his technical support.

REFERENCES

- [1] M. A. Hsieh, V. Kumar, and L. Chaimowicz, "Decentralized controllers for shape generation with robotic swarms," vol. 26, pp. 691–701, 2008.
- [2] A. Galstyan, T. Hogg, and K. Lerman, "Modeling and mathematical analysis of swarms of microscopic robots," in *Swarm Intelligence Symposium, 2005. SIS 2005. Proceedings 2005 IEEE*, june 2005, pp. 201–208.

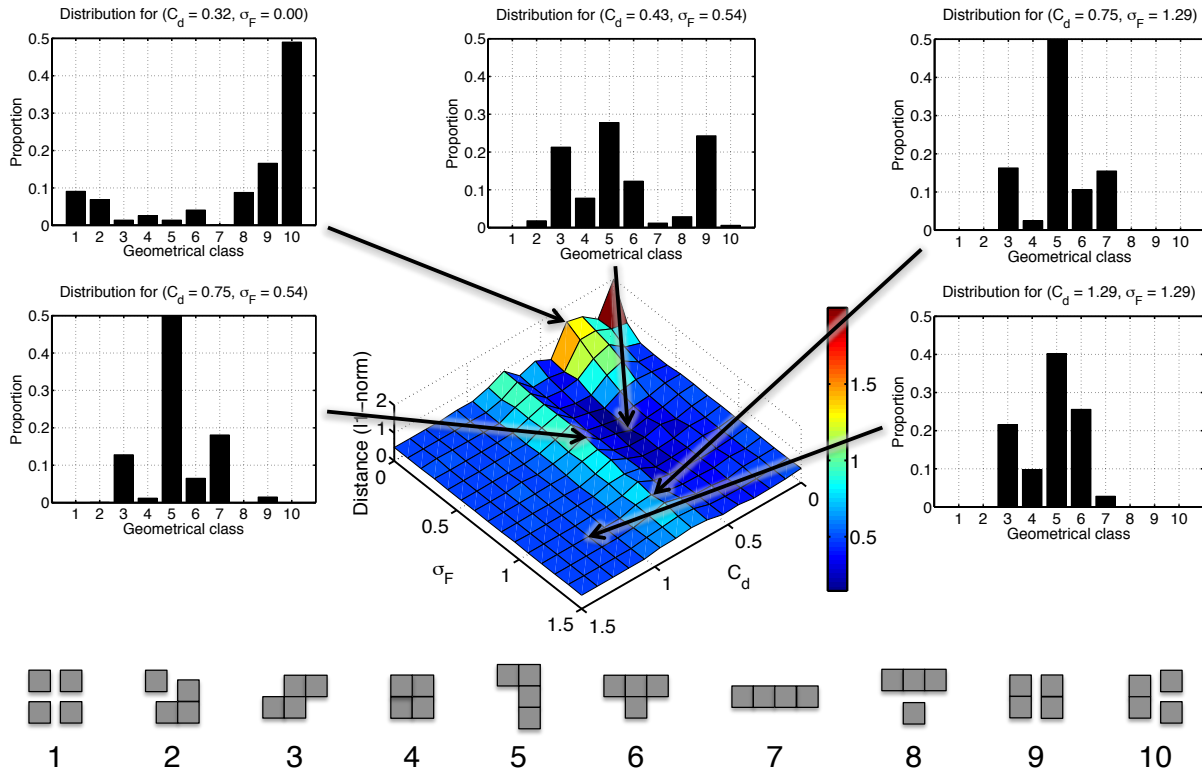


Fig. 11: *Top*: Plot of the distance (as defined in Eq. 5) between the reference distribution (i.e., the distribution yielded by $C_d = 0.45$ and $\sigma_F = 0.55$ mN) and distributions obtained with parameters ranging between 0 and 1.5 for the drag coefficient C_d , and 0 mN and 1.5 mN for the standard deviation σ_F of the stochastic force. *Bottom*: Sketch of the 10 classes of self-assembled geometries discovered during the simulation.

- [3] G. M. Whitesides and B. A. Grzybowski, "Self-assembly at all scales," *Science*, vol. 295, no. 5564, pp. 2418–21, Mar. 2002.
- [4] M. Mastrangeli, S. Abbasi, C. Varel, C. V. Hoof, J.-P. Celis, and K. F. Böhringer, "Self-assembly from milli- to nanoscales: Methods and applications," *J. Micromech. Microeng.*, vol. 19, p. 083001, 2009.
- [5] B. A. Grzybowski, C. E. Wilmer, J. Kim, K. P. Browne, and K. J. M. Bishop, "Self-assembly: from crystals to cells," *Soft Matter*, vol. 5, pp. 1110–1128, 2009.
- [6] H. Wei, Y. Cai, H. Li, D. Li, and T. Wang, "Sambot: A self-assembly modular robot for swarm robot," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, vol. 67, no. 37. IEEE, 2010, pp. 66–71.
- [7] E. Klavins, "Programmable self-assembly," *IEEE Control Systems Magazine*, vol. 27, no. 4, pp. 43–56, 2007.
- [8] W. Evans, G. Mermoud, and A. Martinoli, "Comparing and modeling distributed control strategies for miniature self-assembling robots," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, V. Kumar, Ed. Anchorage, Alaska, USA: IEEE, 2010, pp. 1438–1445.
- [9] J. Neubert, A. Cantwell, S. Constantin, M. Kalontarov, D. Erickson, and H. Lipson, "A robotic module for stochastic fluidic assembly of 3D self-reconfiguring structures," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, vol. 67. IEEE, 2010, pp. 2479–2484.
- [10] S. Miyashita, M. Kessler, and M. Lungarella, "How morphology affects self-assembly in a stochastic modular robot," *Proc. of the 2008 IEEE Int. Conf. on Robotics and Automation (ICRA 2008)*, pp. 3533–3538, 2008.
- [11] K. Gilpin, A. Knaian, and D. Rus, "Robot pebbles: One centimeter modules for programmable matter through self-disassembly," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*. IEEE, 2010, pp. 2485–2492.
- [12] M. Tolley and H. Lipson, "Fluidic manipulation for scalable stochastic 3D assembly of modular robots," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*. IEEE, 2010, pp. 2473–2478.
- [13] M. Mastrangeli, G. Mermoud, and A. Martinoli, "Modeling self-assembly across scales: The unifying perspective of smart minimal particles," *Micromachines*, vol. 2, no. 2, pp. 82–115, 2011.
- [14] N. Napp, D. Thorsley, and E. Klavins, "Hidden Markov Models for non-well-mixed reaction networks," in *American Control Conference (ACC'09)*. IEEE, 2009, pp. 737–744.
- [15] T. Lochmatter, P. Roduit, C. Cianci, N. Correll, J. Jacot, and A. Martinoli, "Swistrack-a flexible open source tracking software for multi-agent systems," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*. Nice, France: IEEE, 2008, pp. 4004–4010.
- [16] O. Michel, "Webots: Professional Mobile Robot Simulation," *Advanced Robotic Systems*, vol. 1, no. 1, pp. 39–42, 2004.
- [17] J. Abbott, Z. Nagy, F. Beyeler, and B. Nelson, "Robotics in the small, part I: Microbotics," *IEEE Robotics & Automation Magazine*, vol. 14, no. 2, pp. 92–103, 2007.
- [18] P. Roduit, "Trajectory Analysis using Point Distribution Models: Algorithms, Performance Evaluation, and Experimental Validation using Mobile Robots," Ph.D. dissertation, École Polytechnique Fédérale de Lausanne, 2009.
- [19] G. Fasano and A. Franceschini, "A multidimensional version of the Kolmogorov-Smirnov test," *Monthly Notices of the Royal Astronomical Society*, vol. 225, pp. 155–170, 1987.
- [20] R. Lopes, I. Reid, and P. Hobson, "The two-dimensional Kolmogorov-Smirnov test," in *XI International Workshop on Advanced Computing and Analysis Techniques in Physics Research, Nikhef, Amsterdam, the Netherlands, April 23-27, 2007*, vol. 335, no. 1, Sept. 2007, pp. 73–83.
- [21] J. Pugh, A. Martinoli, and Y. Zhang, "Particle swarm optimization for unsupervised robotic learning," in *Proc. of the 2005 IEEE Swarm Intelligence Symposium (SIS)*. IEEE, 2005, pp. 92–99.